

### **AMENDMENTS TO THE SPECIFICATION**

**Please replace the paragraph starting at page 1, line 18 with the following**

To execute a program written in the Java-Java™ programming language, the program source file is compiled and a class file, containing virtual machine instructions, is produced. The class file is then executed by a Java-Java™ Virtual Machine (hereinafter JVM). In general, the JVM is an interpreter that decodes and executes the virtual machine instructions contained in the class file. These virtual machine instructions are referred to as bytecodes. In addition to the bytecodes, the class file includes other information utilized and/or operated on by the JVM. For example, the class file includes a number to identify the class file format and a constant pool table to identify constants that are referenced within the class file.

**Please replace the paragraph starting at page 2, line 1 with the following**

The Java-Java™ programming language is an interpreted language. Generally, an interpreted language is a language in which programs are translated and executed one statement at a time. As an example, in the Java-Java™ environment, the virtual machine instructions contained in a class file are interpreted by the JVM, translated on-the-fly into native machine code, and then executed.

**Please replace the paragraph starting at page 2, line 6 with the following**

A disadvantage of interpretive languages involves speed of execution. Being interpreted, the Java-Java™ programming language can be slower than other languages implemented with compilers when each is executing the same algorithms. However, as an interpreted language, the Java-Java™ language is able to provide a more flexible run time environment in which classes and dynamic constructs contained in the Java-Java™ programming language can be